Terms used **reorder buffer nop** Found **19** of **166,357**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 19 of 19

Relevance scale

1 Dynamic dead-instruction detection and elimination

J. Adam Butts, Guri Sohi

October 2002 **ACM SIGOPS Operating Systems Review , ACM SIGPLAN Notices , ACM SIGARCH Computer Architecture News , Proceedings of the 10th international conference on Architectural support for programming languages and operating systems ASPLOS-X**, Volume 36 , 37 , 30 Issue 5 , 10 , 5

Publisher: ACM Press

Full text available: pdf(1.50 MB) Additional Information: full citation, abstract, references, citings

We observe a non-negligible fraction--3 to 16% in our benchmarks--of *dynamically dead instructions,* dynamic instruction instances that generate unused results. The majority of these instructions arise from static instructions that also produce useful results. We find that compiler optimization (specifically instruction scheduling) creates a significant portion of these *partially dead* static instructions. We show that most of the dynamically instructions arise from a small set of st ...

2 Boosting beyond static scheduling in a superscalar processor

Michael D. Smith, Monica S. Lam, Mark A. Horowitz

May 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture ISCA '90**, Volume 18 Issue 3a

Publisher: ACM Press

Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a superscalar processor that combines the best qualities of static and dynamic instruction scheduling to increase the performance of non-numerical applications. The architecture performs all instruction scheduling statically to take advantage of the compiler's ability to efficiently schedule operations across many basic blocks. Since the conditional branches in non-numerical code are highly data dependent, the architecture introduces the concept of boosted

3 VHC: Quickly Building an Optimizer for Complex Embedded Architectures

Michael Dupré, Nathalie Drach, Olivier Temam

March 2004 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '04**

Publisher: IEEE Computer Society

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, index terms

To meet the high demand for powerful embedded processors,VLIW architectures are